HL Klemove

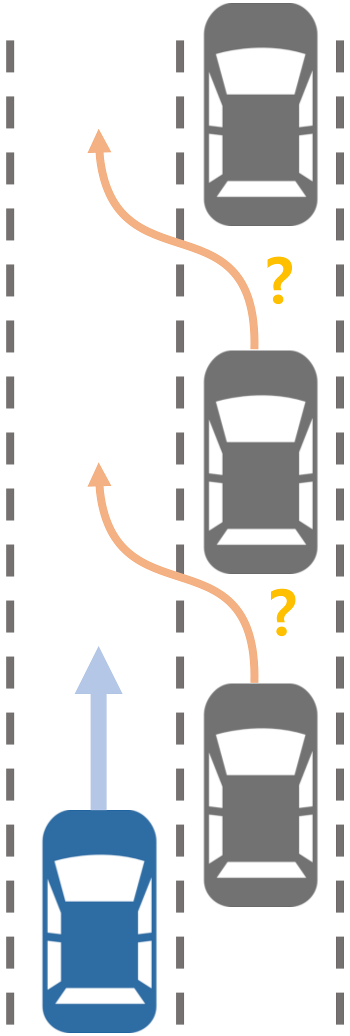
Prediction

Lecturer : Seungmok Song

Contents

Introduction

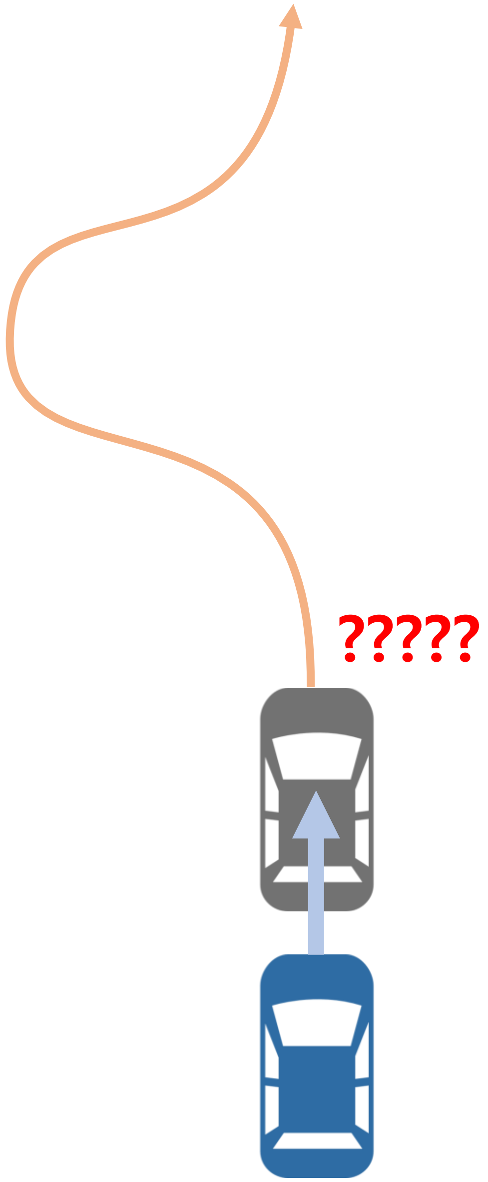
- Prediction
  - 초보 운전 vs 숙련된 운전자



Introduction

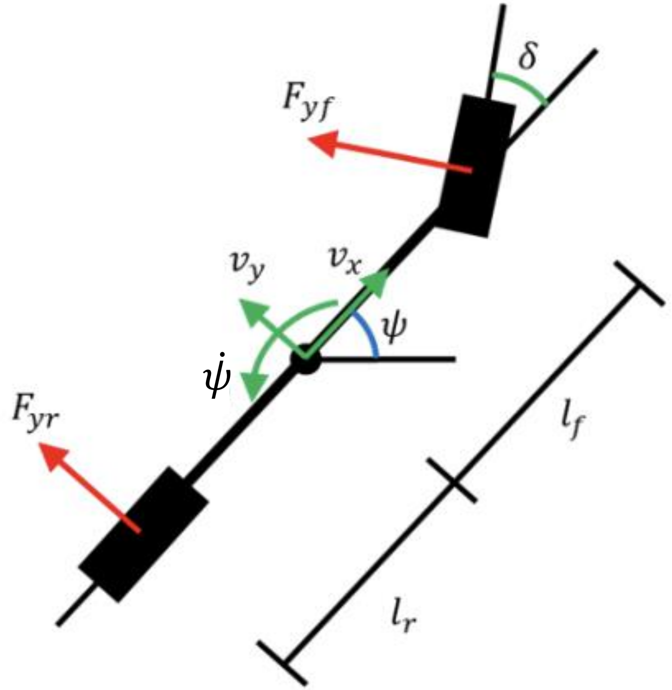
- Prediction
  - 초보 운전 vs 숙련된 운전자

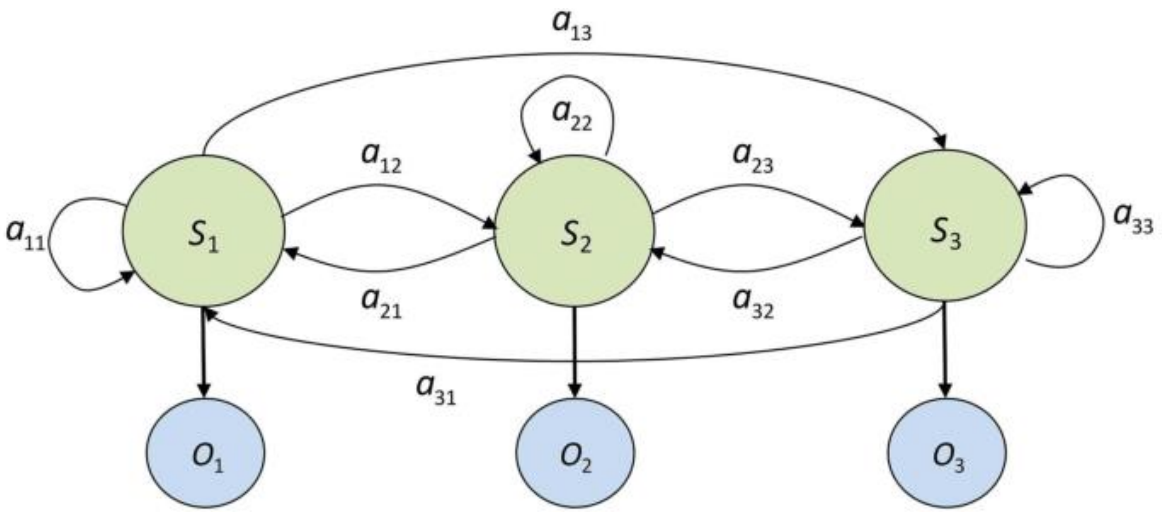
결국 상대방의 차량 움직임을 예측하고 싶은 것



Introduction

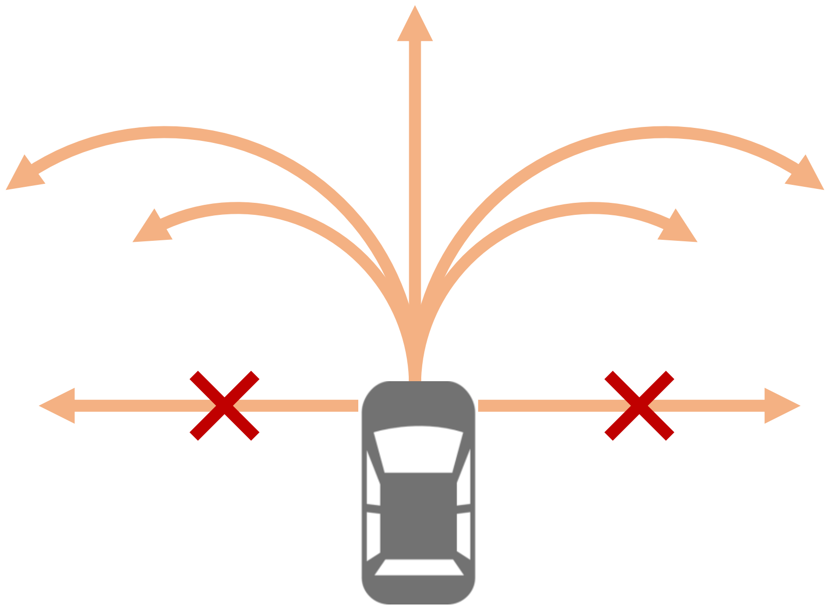
- Model based prediction
  - Physics based prediction
  - Maneuver based prediction



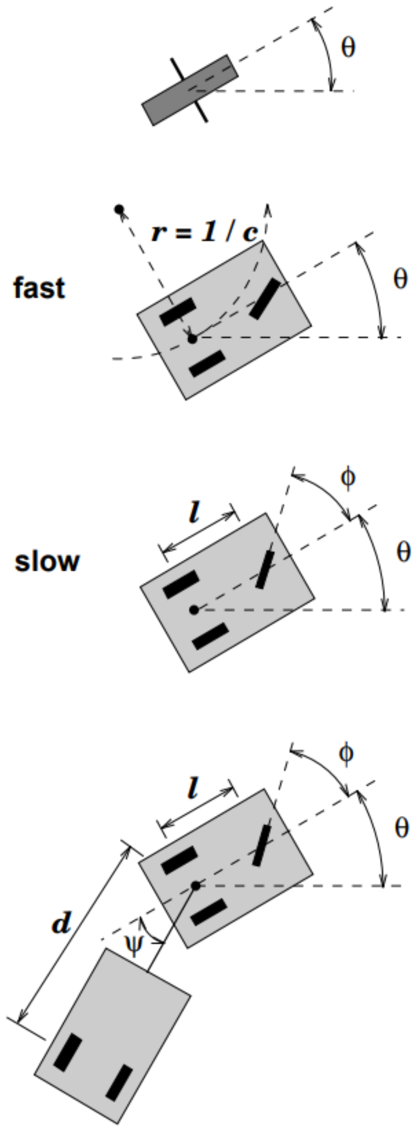


Physics based prediction

- Introduction
  - 자동차는 정해진 물리 법칙 상에서 움직임
  - 짧은 시간 예측에 유리
  - 주변 환경을 고려하지 않음
  - CV / CA / CTRV / CTRA model



Nonholonomic Systems



Unicycle

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} v \\ v_\theta \end{pmatrix}$$

Car with fast steering

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} v \\ v\,c \end{pmatrix}$$

Car with slow steering

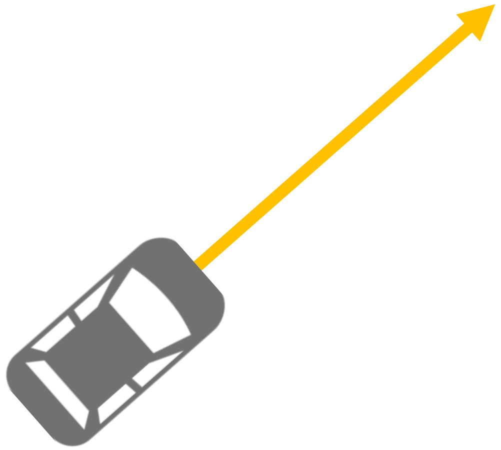
$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{pmatrix} x \\ y \\ \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \tan(\phi)/l & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} v \\ v_\phi \end{pmatrix}$$

Car with trailer

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{pmatrix} x \\ y \\ \theta \\ \psi \\ \phi \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \tan(\phi)/l & 0 \\ \sin(\psi)/d & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} v \\ v_\phi \end{pmatrix}$$

Physics based prediction

- Constant velocity model (CV model)
  - 현재의 주행 속도를 유지한다는 가정
  - 가장 간단한 모델



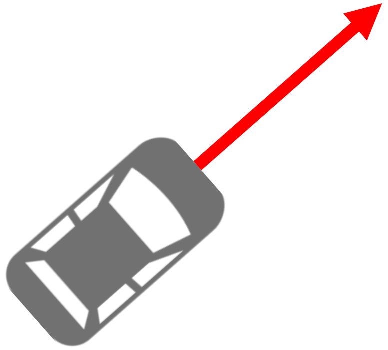
State : $X = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}$

$$\begin{cases} x_{k+1} = x_k + v_{x,k}\Delta t \\ y_{k+1} = y_k + v_{y,k}\Delta t \\ v_{x,k+1} = v_{x,k} \\ v_{y,k+1} = v_{y,k} \end{cases}$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ v_{x,k+1} \\ v_{y,k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{bmatrix}$$

Physics based prediction

- Constant acceleration model (CA model)
  - 현재의 가속도를 유지한다는 가정
  - 주의해야 할 점..?

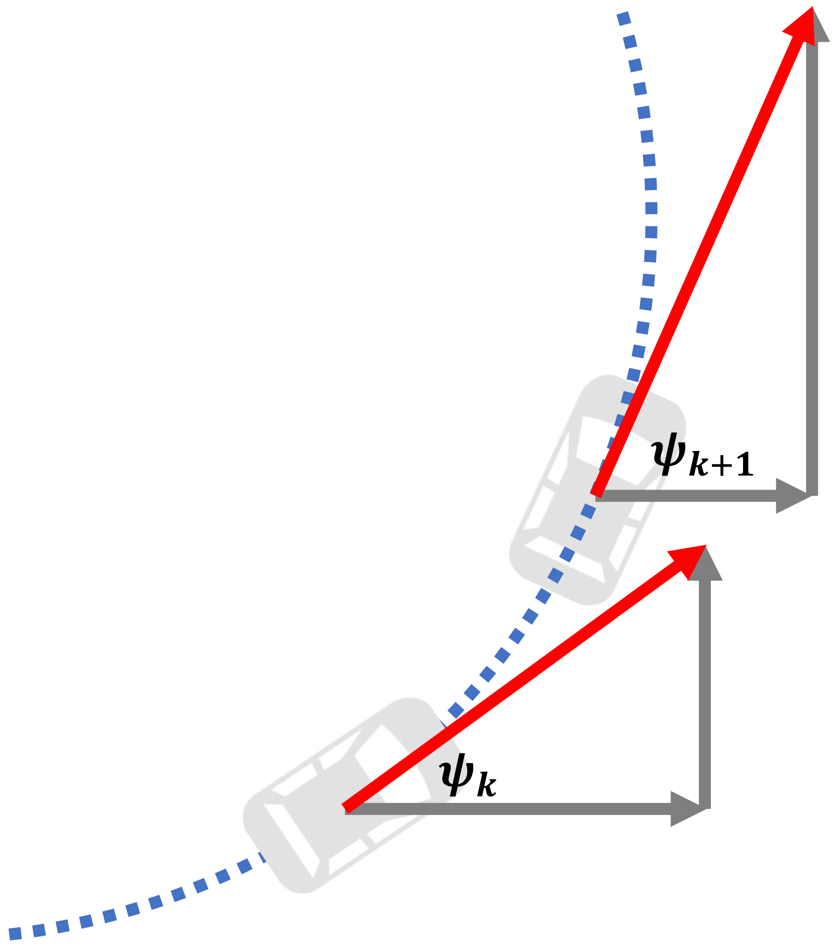


$$State: \; X = \begin{bmatrix} x \\ v \\ a \end{bmatrix} \qquad \begin{cases} x_{k+1} = x_k + v_k \Delta t + \frac{1}{2} a_k \Delta t^2 \\ v_{k+1} = v_k + a_k \Delta t \\ a_{k+1} = a_k \end{cases}$$

$$\begin{bmatrix} \boldsymbol{x_{k+1}} \\ \boldsymbol{v_{k+1}} \\ \boldsymbol{a_{k+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{\Delta t} & \boldsymbol{\Delta t^2} \\ \mathbf{0} & \mathbf{1} & \boldsymbol{\Delta t} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x_k} \\ \boldsymbol{v_k} \\ \boldsymbol{a_k} \end{bmatrix}$$

Physics based prediction

- Constant turn rate & velocity model (CTRV model)
  - 현재의 yaw rate(turn rate) 와 속도를 유지한다는 가정
  - If (yaw rate << 1) ?



$$State: X = \begin{bmatrix} x \\ y \\ v \\ \psi \\ \dot{\psi} \end{bmatrix}$$

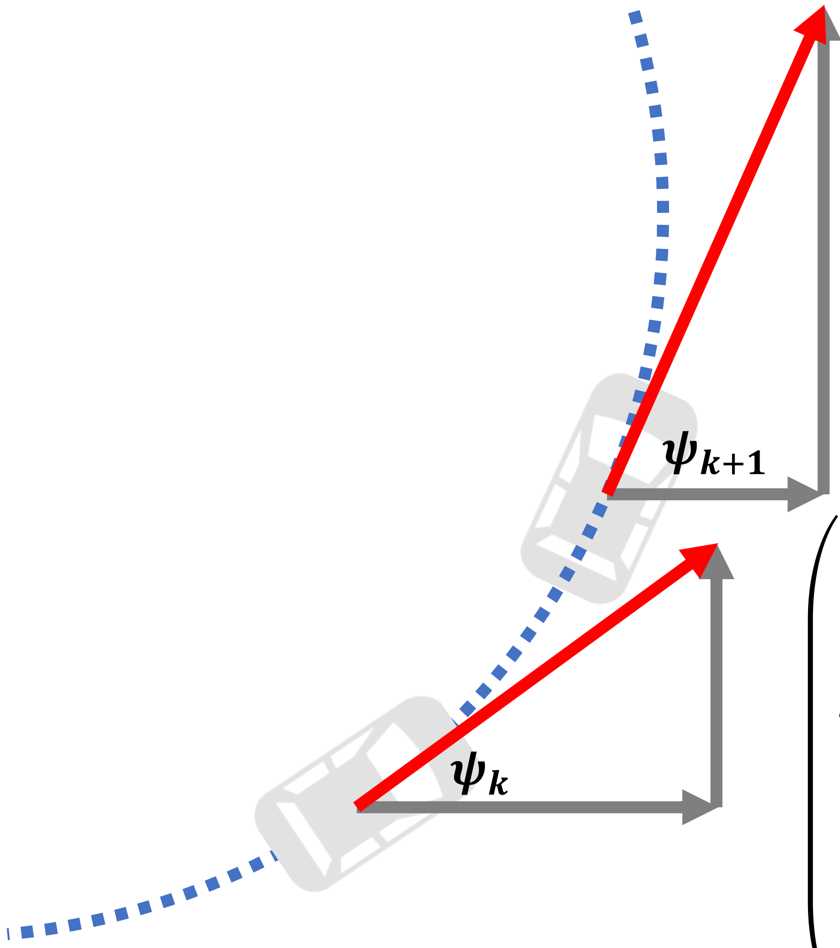
$$\begin{cases} x_{k+1} = x_k + \int_0^{\Delta t} v_{k,\tau}\cos(\psi_{k,\tau})d\tau \\ y_{k+1} = y_k + \int_0^{\Delta t} v_{k,\tau}\sin(\psi_{k,\tau})d\tau \\ v_{k,\tau} = v_k(const) \\ \psi_{k,\tau} = \psi_k + \dot{\psi}_k\tau \end{cases} \Rightarrow \begin{aligned} x_{k+1} &= x_k + \int_0^{\Delta t} v_k\cos(\psi_k + \dot{\psi}_k\tau)d\tau \\ y_{k+1} &= y_k + \int_0^{\Delta t} v_k\sin(\psi_k + \dot{\psi}_k\tau)d\tau \end{aligned}$$

$$\begin{aligned} \boldsymbol{x_{k+1}} &= \boldsymbol{x_k + \frac{v_k}{\dot{\psi}_k}\left(\sin(\psi_k + \dot{\psi}_k\Delta t) - \sin(\psi_k)\right)} \\ \boldsymbol{y_{k+1}} &= \boldsymbol{y_k + \frac{v_k}{\dot{\psi}_k}\left(-\cos(\psi_k + \dot{\psi}_k\Delta t) + \cos(\psi_k)\right)} \\ \boldsymbol{v_{k+1}} &= \boldsymbol{v_k} \\ \boldsymbol{\psi_{k+1}} &= \boldsymbol{\psi_k + \dot{\psi}_k\Delta t} \\ \boldsymbol{\dot{\psi}_{k+1}} &= \boldsymbol{\dot{\psi}_k} \end{aligned}$$

Physics based prediction

- Constant turn rate & acceleration model (CTRA model)
    - 현재의 yaw rate(turn rate) 와 가속도를 유지한다는 가정

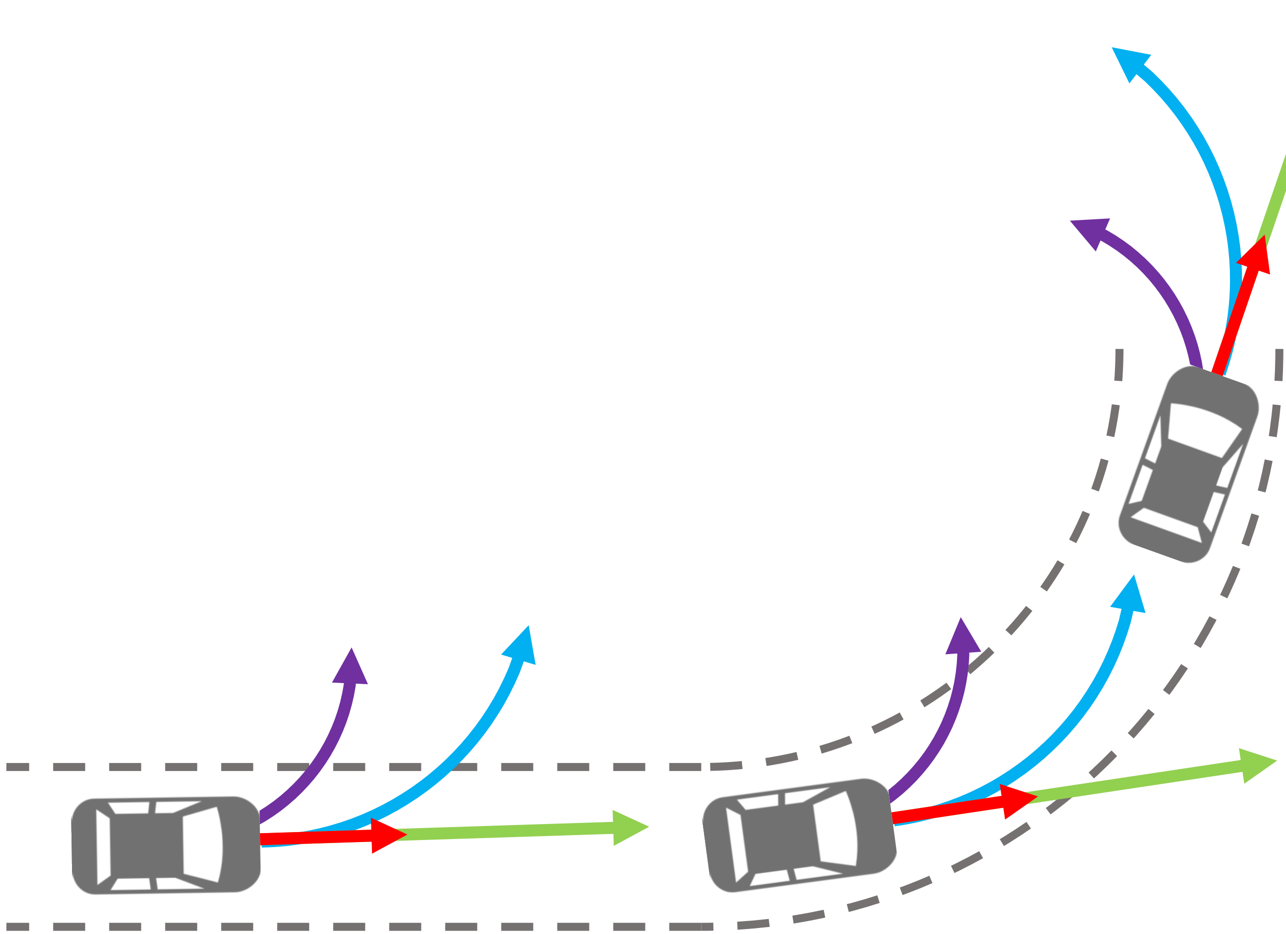
$$X = \begin{bmatrix} x \\ y \\ v \\ a \\ \psi \\ \dot{\psi} \end{bmatrix} \qquad \begin{cases} v_{k+1} = v_k + a_k \Delta t \\ a_{k+1} = a_k \\ \psi_{k+1} = \psi_k + \dot{\psi}_k \Delta t \\ \dot{\psi}_{k+1} = \dot{\psi}_k \\ x_{k+1} = x_k + \int_0^{\Delta t} v_{k,\tau} \cos(\psi_{k,\tau}) d\tau \\ y_{k+1} = y_k + \int_0^{\Delta t} v_{k,\tau} \sin(\psi_{k,\tau}) d\tau \end{cases}$$



$$\begin{cases} \boldsymbol{x_{k+1} = x_k + \frac{v_k}{\dot{\psi}_k}(\sin(\psi_k + \dot{\psi}_k \Delta t) - \sin(\psi_k)) + \frac{a_k}{\dot{\psi}_k^2}(\cos(\psi_k + \dot{\psi}_k \Delta t) + \dot{\psi}_k \Delta t \sin(\psi_k + \dot{\psi}_k \Delta t) - \cos(\psi} \\ \boldsymbol{y_{k+1} = y_k - \frac{v_k}{\dot{\psi}_k}(\cos(\psi_k + \dot{\psi}_k \Delta t) - \cos(\psi_k)) + \frac{a_k}{\dot{\psi}_k^2}(\sin(\psi_k + \dot{\psi}_k \Delta t) - \dot{\psi}_k \Delta t \cos(\psi_k + \dot{\psi}_k \Delta t) - \sin(\psi} \\ \boldsymbol{v_{k+1} = v_k + a_k \Delta t} \\ \boldsymbol{a_{k+1} = a_k} \\ \boldsymbol{\psi_{k+1} = \psi_k + \dot{\psi}_k \Delta t} \\ \boldsymbol{\dot{\psi}_{k+1} = \dot{\psi}_k} \end{cases}$$

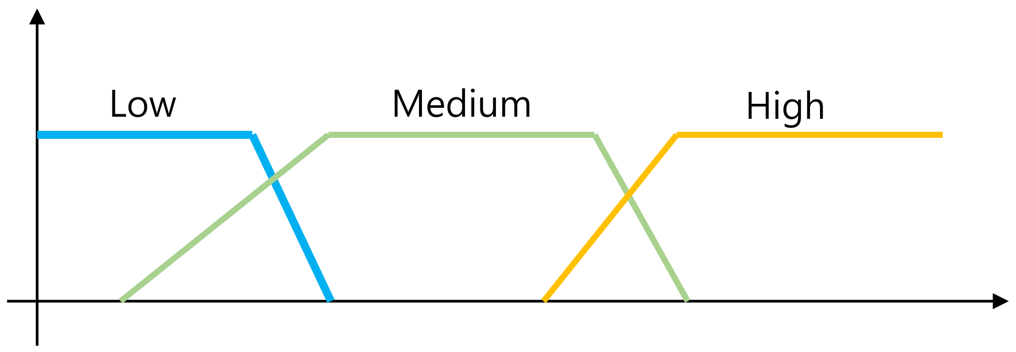
Physics based prediction

- Which model is the best?
  - CV, CA, CTRV, CTRA

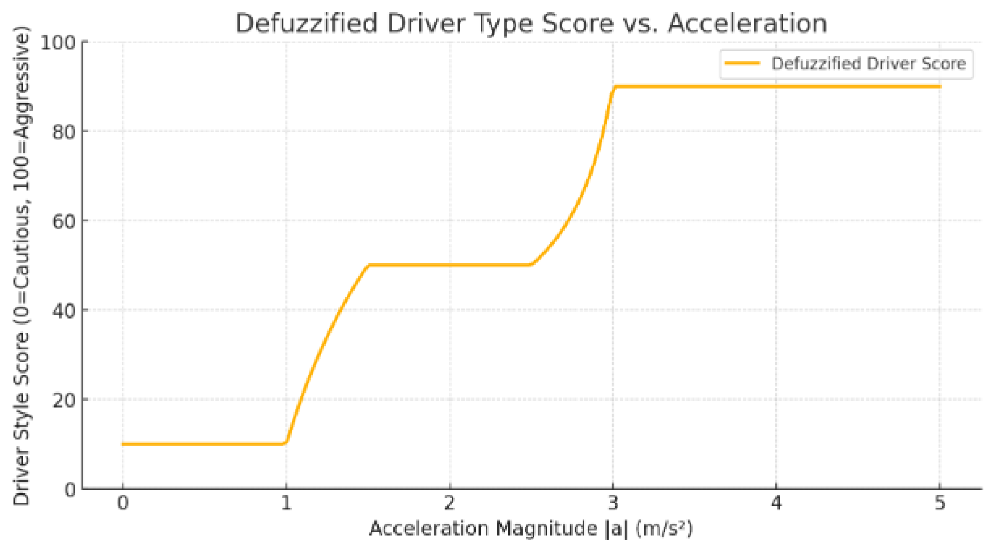


Physics based prediction

- Fuzzy theory
  - Deal with ambiguous and uncertain situation
  - Fuzzification(퍼지화)
    - Membership function
      - Input : Average absolute value of acceleration
      - Input label : Low / Medium / High
      - Output label : Cautious / Normal / Aggressive
      - Form : triangle / sigmoid

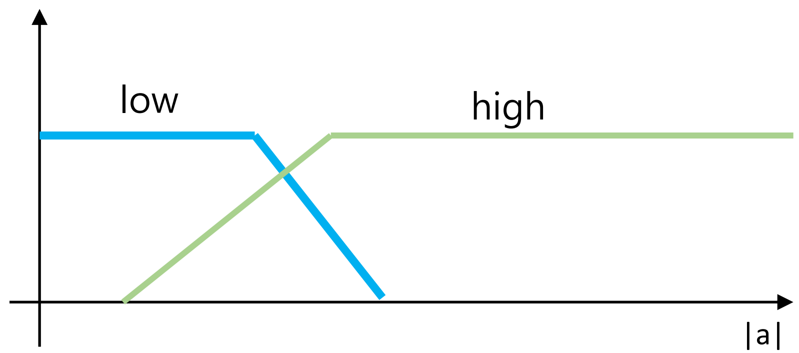


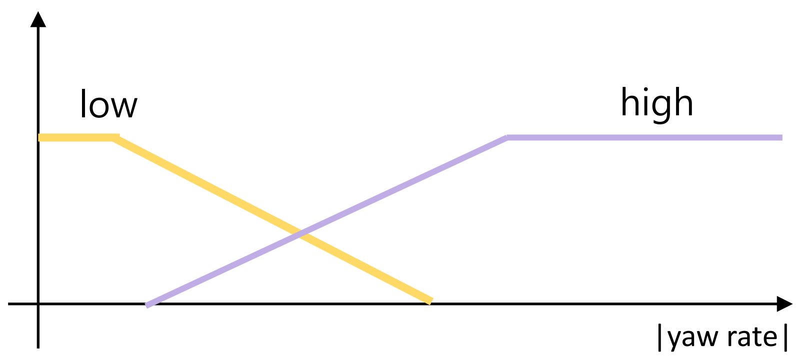
  - Fuzzy rules (퍼지 규칙)
    - If |a| is low, the driver is cautious for 80% chance, Normal for 20% chance
    - If |a| is medium, the driver is Normal for 100% chance
    - If |a| is high, the driver is Aggressive for 100% chance
  - Defuzzification (디퍼지화)
    - If the driver is cautious driver; the score is 0
    - If the driver is normal driver; the score is 50
    - If the driver is aggressive driver; the score is 100



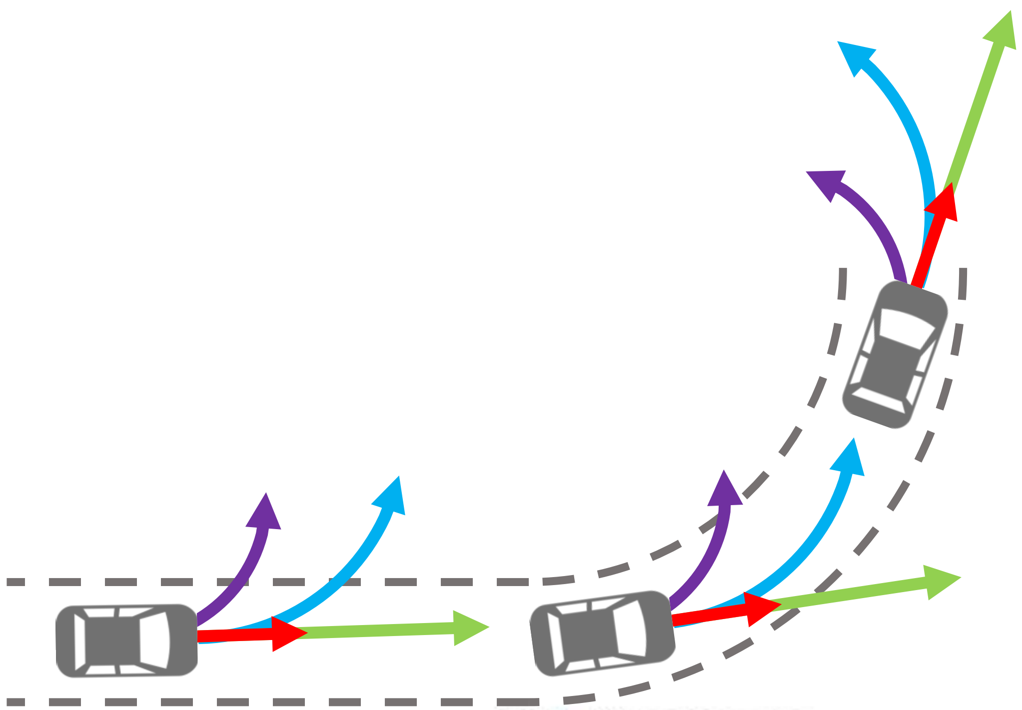
Physics based prediction

- Fuzzy theory
  - Fuzzification(퍼지화)
    - Membership function
      - Input : longitudinal acceleration & yaw rate
      - Input label : Low / High acceleration & yaw rate
      - Output label : accelerating / turning
      - Form : triangle



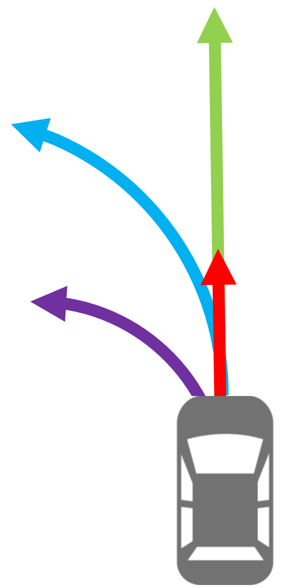


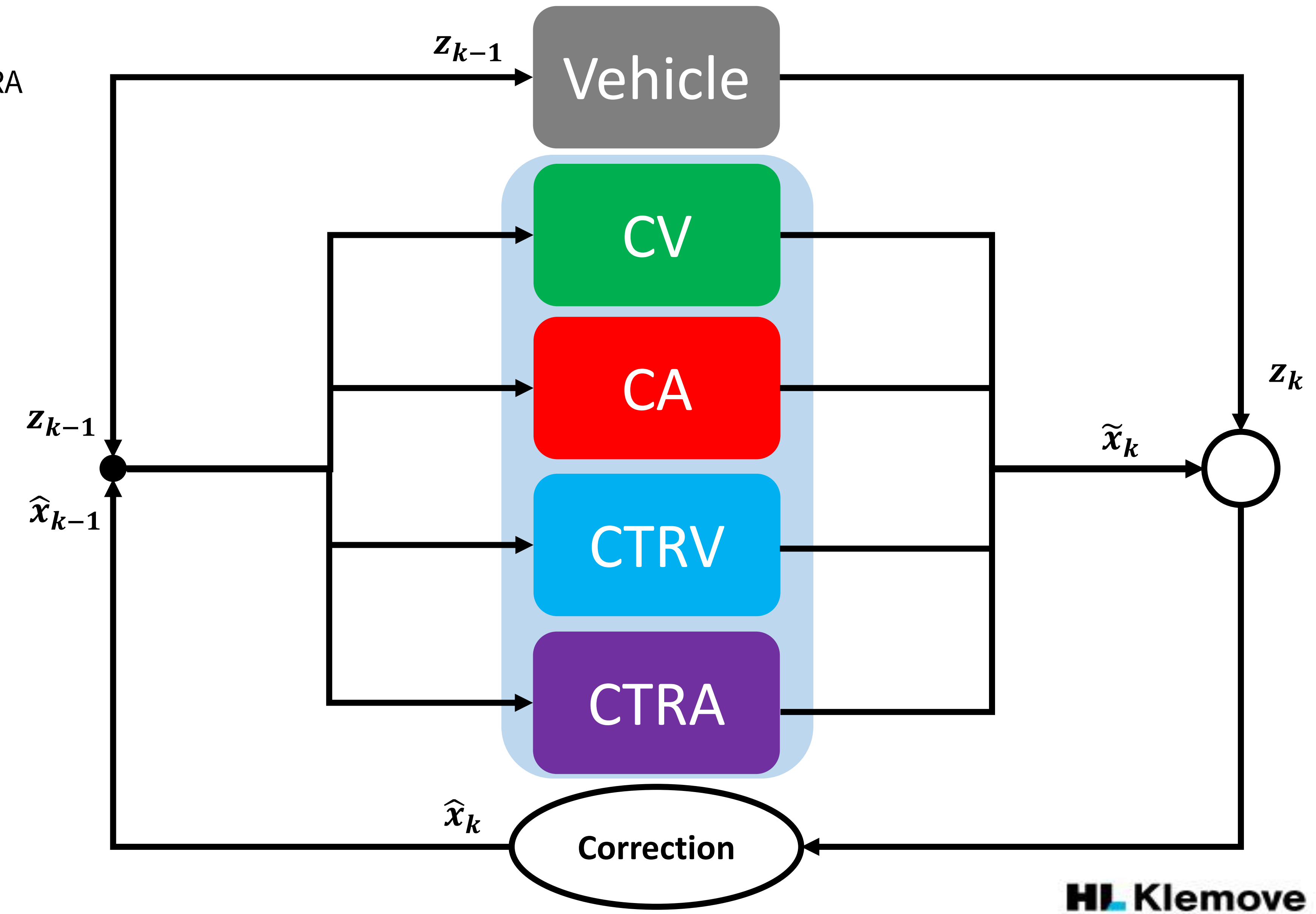
  - Fuzzy rules (퍼지 규칙)
    - If acceleration is low, the the vehicle will follow (CV[50%], CA[0%] , CTRV[50%], CTRA[0%]) model
    - If acceleration is high, the the vehicle will follow (CV[0%], CA[70%] , CTRV[0%], CTRA[30%]) model
    - If yaw rate is low, the the vehicle will follow (CV[40%], CA[60%] , CTRV[0%], CTRA[0%]) model
    - If yaw rate is high, the the vehicle will follow (CV[0%], CA[0%] , CTRV[60%], CTRA[40%]) model
    - Aggregate : Weight combination – normalize (Σ(w_case*P_model))
  - Defuzzification (디퍼지화)
    - Weighted average of trajectories from each model; CV, CA, CTRV, CTRA.



Physics based prediction

- Interacting multiple model (IMM)
    - Which model is the best?- CV, CA, CTRV, CTRA





Physics based prediction

- Interacting multiple model (IMM)
  - Basic Kalman filter

① Prediction

$$\tilde{x}_k = A\hat{x}_{k-1} + Bu_k$$

$$\tilde{y}_k = C\tilde{x}_k$$

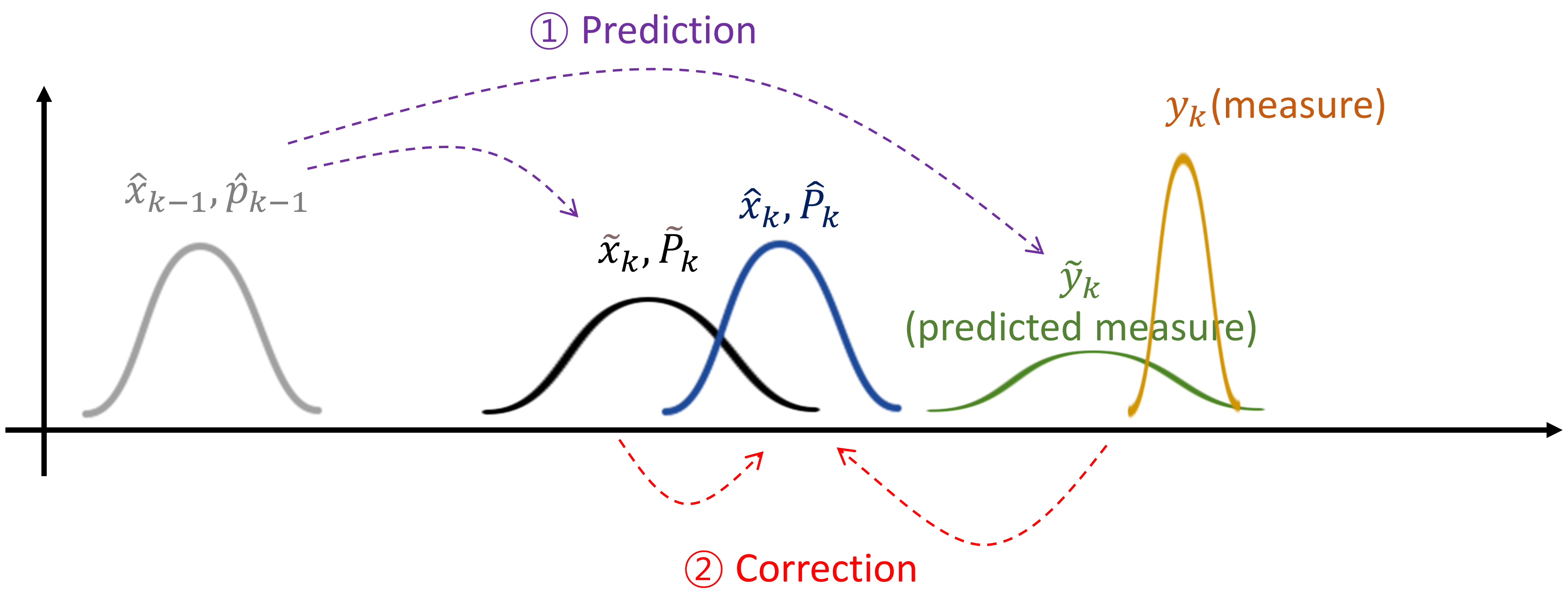
$$\tilde{P}_k = AP_{k-1}A^T + \underline{Q_k}$$

② Correction

$$\hat{x}_k = \tilde{x}_k + \underline{K_k}(y_k - \tilde{y}_k)$$

$$K_k = \frac{\tilde{P}_k C^T}{C\tilde{P}_k C^T + \underline{R_k}}$$

$$P_k = (I - K_k C)\tilde{P}_k$$



K_k(Kalman gain) : 예측된 값에 측정값과의 차이를 얼마나 반영할 것인지에 대한 값

Q_k(Model noise) : Kalman gain 을 크게 하는 요소(Tuning parameter)

R_k(Sensor noise) : Kalman gain 을 작게 하는 요소 (Tuning parameter)

Physics based prediction

- Interacting multiple model (IMM)

① Prediction

$$\tilde{x}_k^i = f^i(\hat{x}_{k-1}^i)$$

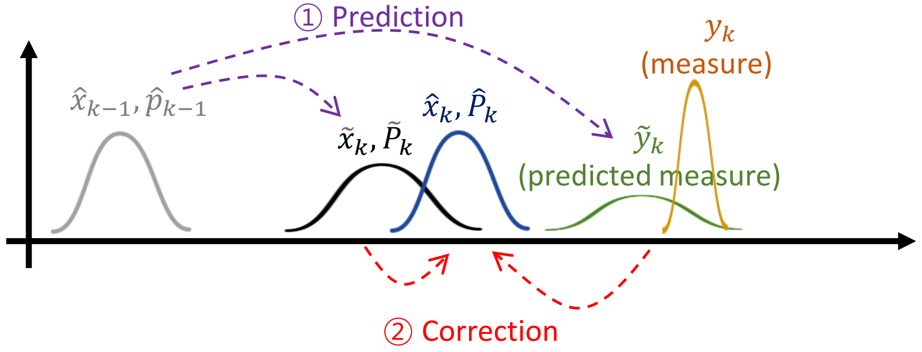
$$\tilde{P}_k^i = F^i P_{k-1}^i F^{iT} + Q^i$$

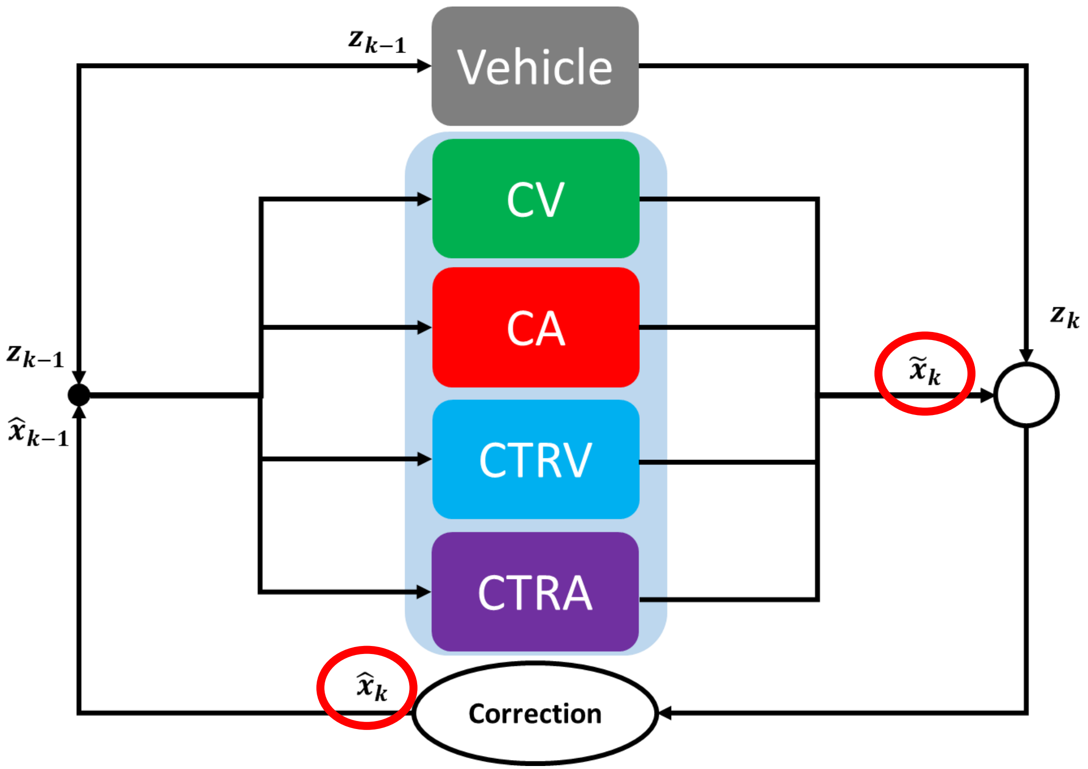
② Correction

$$\hat{x}_k^i = \tilde{x}_k^i + K_k^i\left(z_k^i - h^i(\tilde{x}_k^i)\right)$$

$$K_k^i = \frac{\tilde{P}_k^i H^{iT}}{H^i \tilde{P}_k^i H^{iT} + R^i}$$

$$P_k^i = (I - K_k^i H^i)\tilde{P}_k^i$$





HL Klemove

Physics based prediction

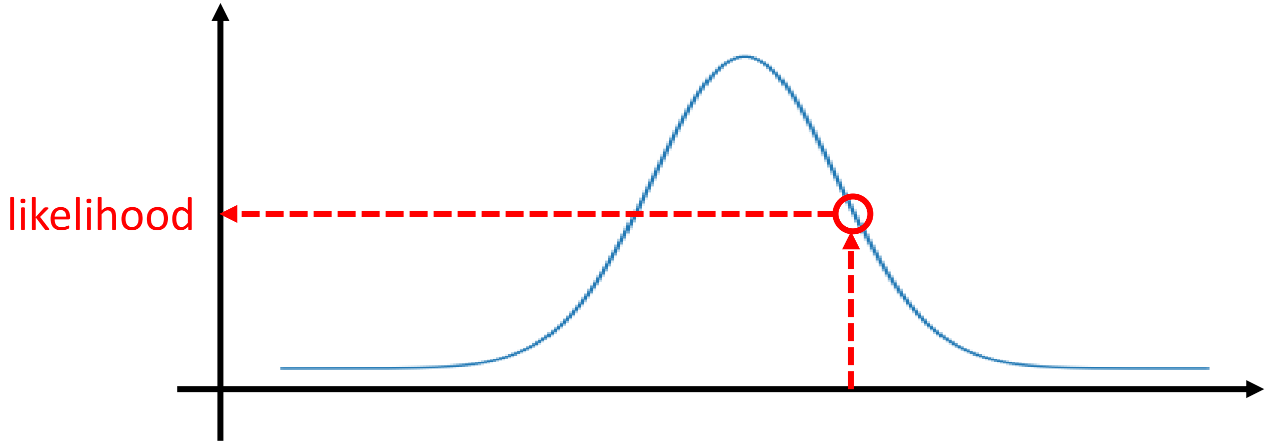
- Interacting multiple model (IMM)

③ Likelihood update

Bayes 정리; $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$, A : 모델 i 가 맞음 / B : 관측 z 발생

$P(B|A)$: Likelihood, 모델 i 가 맞을 때 관측 z 발생할 확률 – from Gaussian distribute model

$$\Lambda_k^i = \frac{1}{\sqrt{(2\pi)^d \left(H^i \tilde{P}_k^i H^{iT} + R^i\right)}} e^{\left(\frac{1}{2}\left(z_k^i - h^i\left(\tilde{x}_k^i\right)\right)\left(H^i \tilde{P}_k^i H^{iT} + R^i\right)^T \left(z_k^i - h^i\left(\tilde{x}_k^i\right)\right)^T\right)}$$



④ Update model

$P(A)$: 모델 i 가 맞을 확률(관측 전)

$$c^i = \sum_j \pi_{ij} \mu_{k-1}^j$$

$P(B)$: 관측 z 발생할 확률

$$c^i = \sum_j c^j \Lambda_k^j$$

$P(A|B)$: 관측 z 발생했을 때 모델 I 가 맞을 확률

$$\mu_k^i = \frac{c^i \Lambda_k^i}{\sum_j c^j \Lambda_k^j}$$

Physics based prediction

- Interacting multiple model (IMM)

⑤ Mixing model

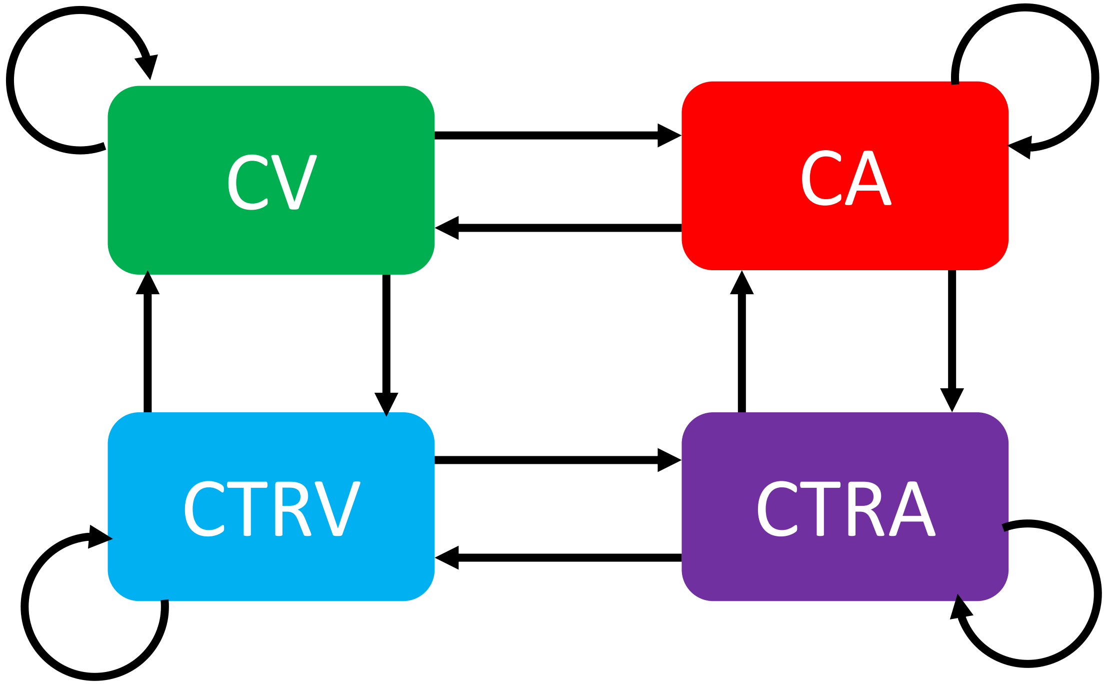
추정값 혼합

$$\hat{x}_k^{i*} = \sum_j \frac{\pi_{ij}\mu_k^j}{c^i}$$

공분산 혼합

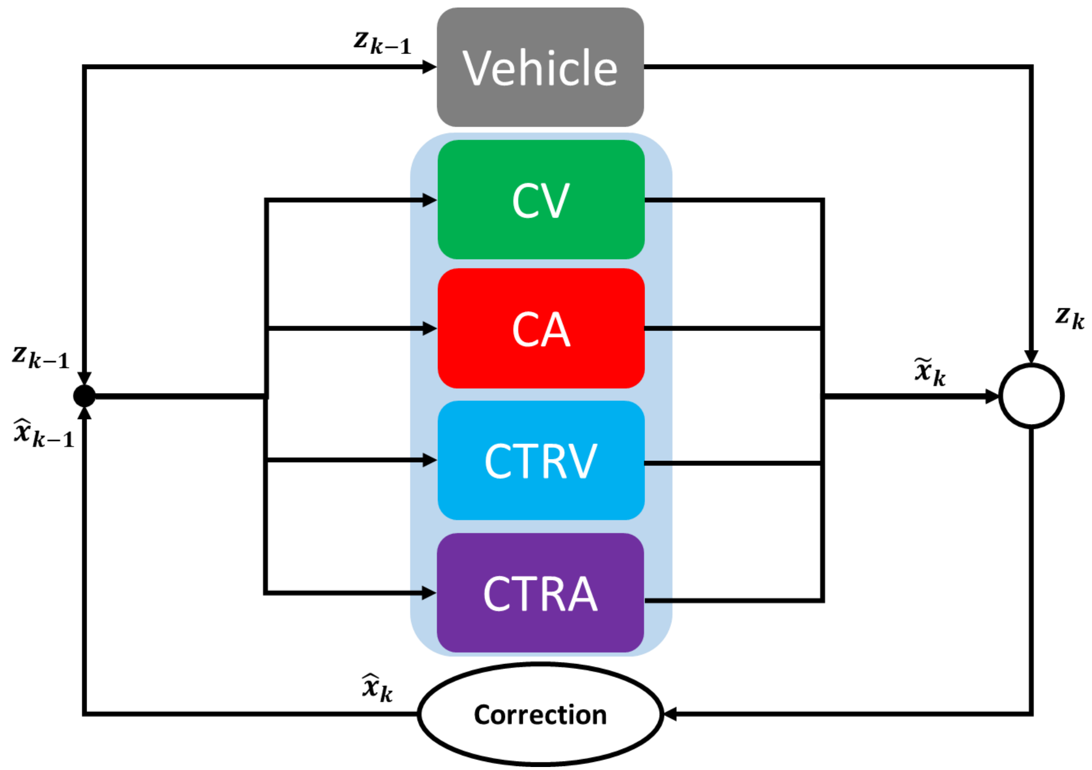
$$P_k^{i*} = \sum_j \frac{\pi_{ij}\mu_k^j}{c^j}\left(P_k^j + \left(\hat{x}_k^j - \hat{x}_j^i\right)\left(\hat{x}_k^j - \hat{x}_j^i\right)^T\right)$$

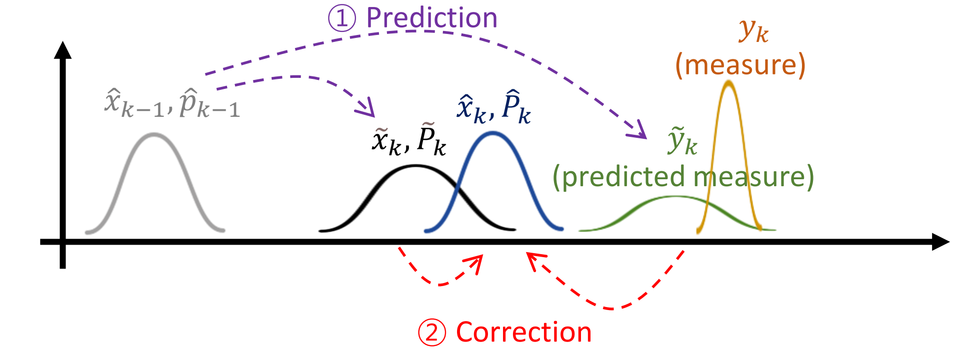
$$\pi_{ij} = P\left(M_i \middle| M_j\right) = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1m} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{m1} & \pi_{m2} & \cdots & \pi_{mm} \end{bmatrix}$$

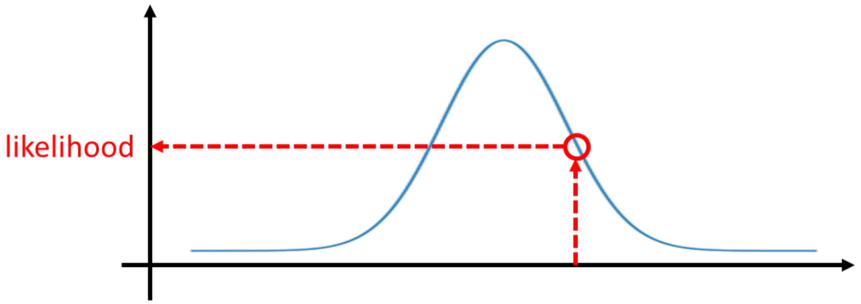


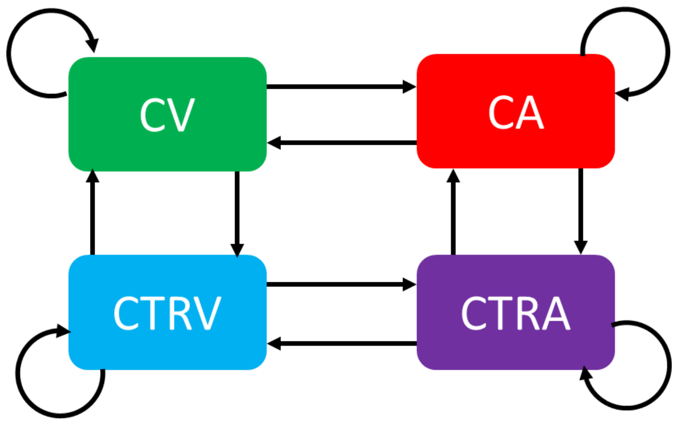
Physics based prediction

- Interacting multiple model (IMM)



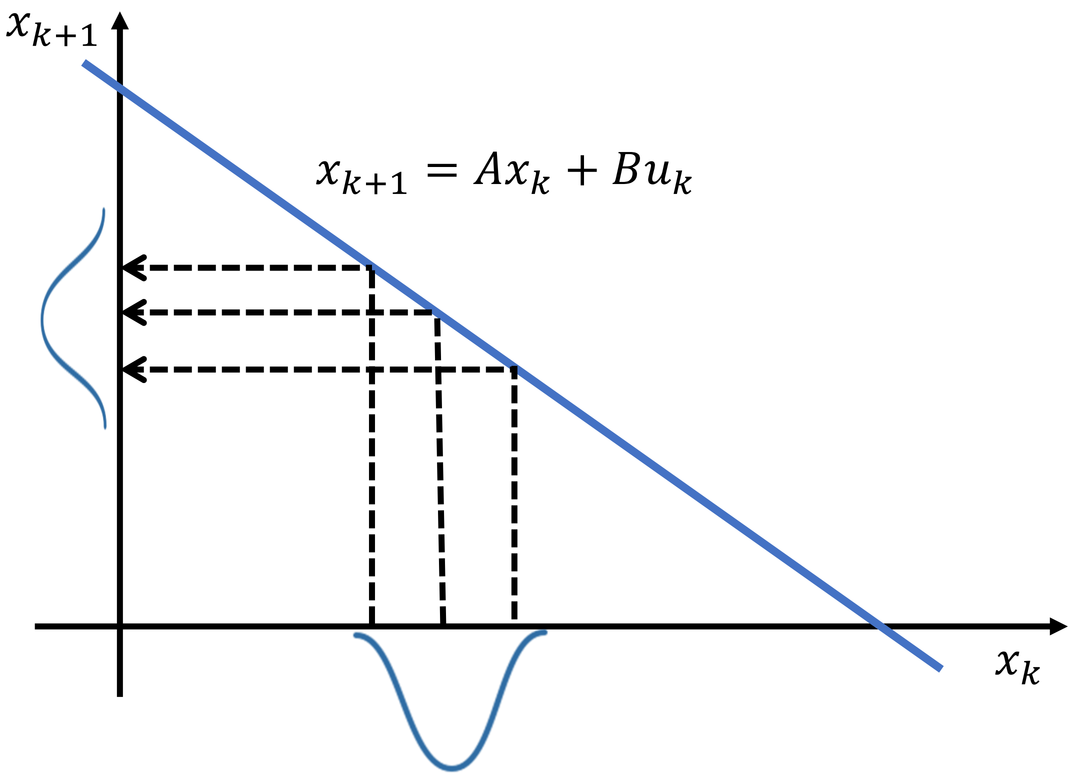


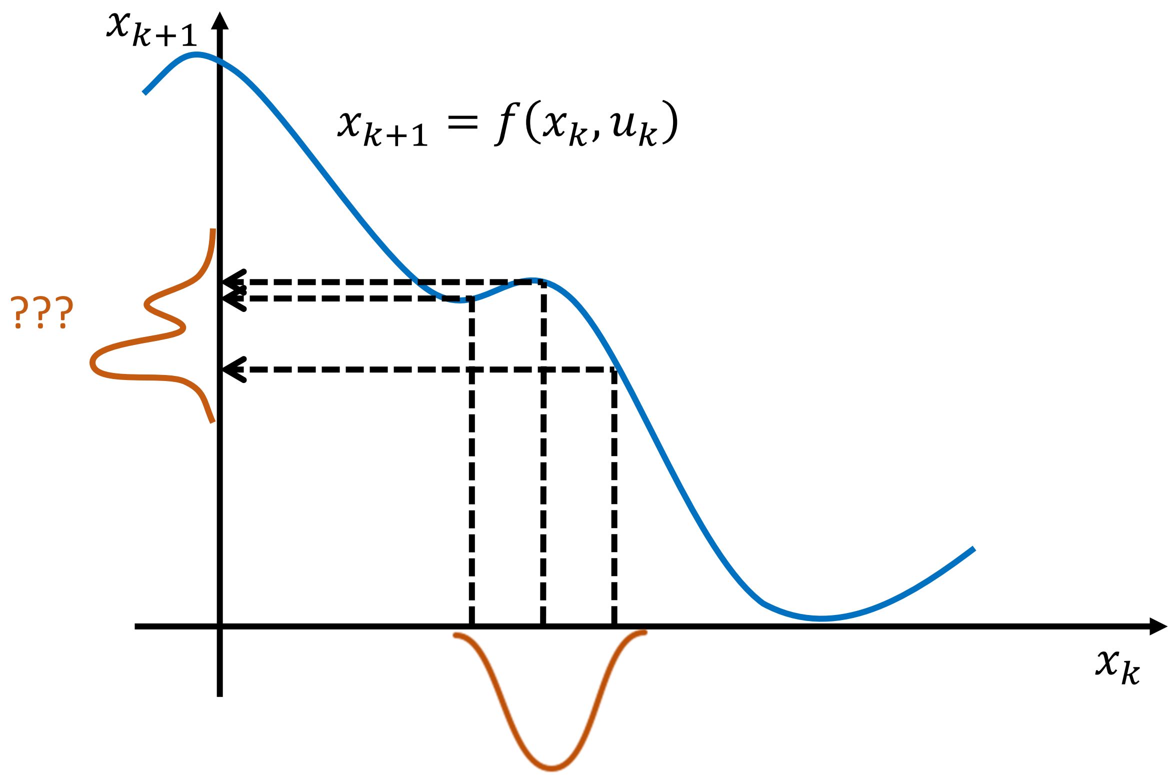




Physics based prediction

- Nonlinear Kalman filter
  - Linear vs Nonlinear model





Physics based prediction

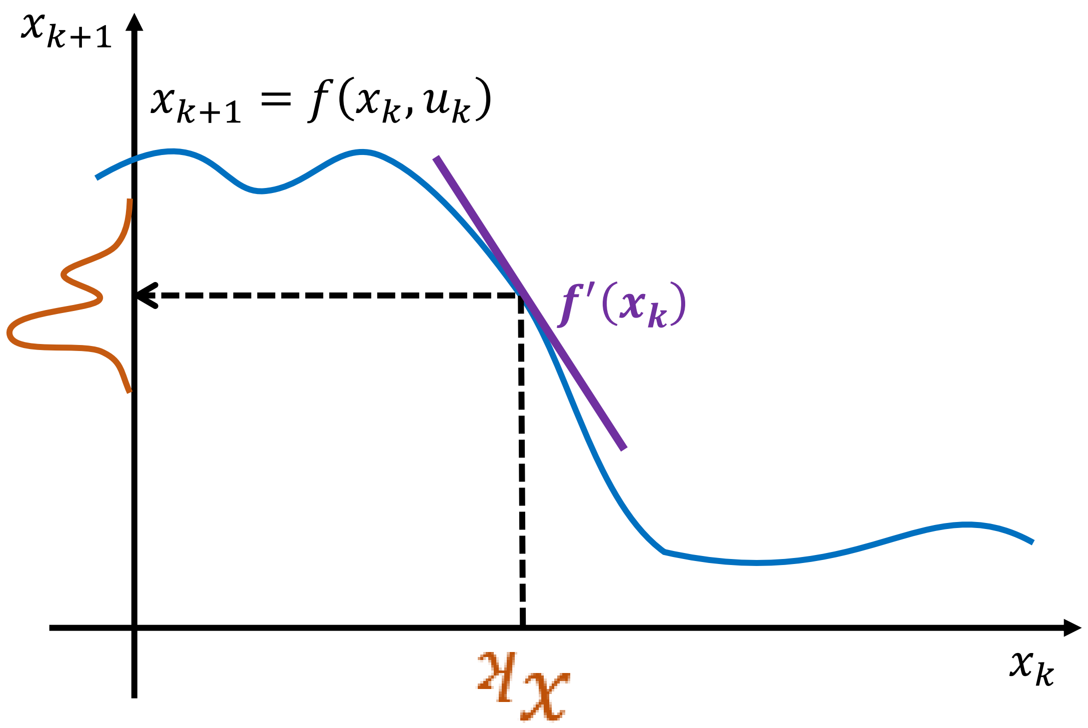
- Nonlinear Kalman filter
  - Idea 1 : Taylor series
    - 특정 함수는 한 점에서의 도함수 값들과 다항식의 무한 합으로 나타낼 수 있음

$$f(x) = \underline{f(a) + f'(a)(x-a)} + \frac{1}{2} f''(a)(x-a)^2 + \ldots$$

$$f(x,y,z) = f(x_i,y_i,z_i) + (x - x_i)\left.\frac{\partial f}{\partial x}\right|_{x_i,y_i,z_i}$$

$$+(y - y_i)\left.\frac{\partial f}{\partial y}\right|_{x_i,y_i,z_i}$$

$$+(z - z_i)\left.\frac{\partial f}{\partial z}\right|_{x_i,y_i,z_i}$$



Physics based prediction

- Nonlinear Kalman filter
  - Example : CTRV Model

$$x_{k+1} = x_k + \frac{v_k}{\dot{\psi}_k}\left(\sin(\psi_k + \dot{\psi}_k \Delta t) - \sin(\psi_k)\right) = \boldsymbol{f_1(x_k, y_k, \dots)}$$

$$y_{k+1} = y_k + \frac{v_k}{\dot{\psi}_k}\left(-\cos(\psi_k + \dot{\psi}_k \Delta t) + \cos(\psi_k)\right) = \boldsymbol{f_2(x_k, y_k, \dots)}$$

$$v_{k+1} = v_k = \boldsymbol{f_3(x_k, y_k, \dots)}$$

$$\psi_{k+1} = \psi_k + \dot{\psi}_k \Delta t = \boldsymbol{f_4(x_k, y_k, \dots)}$$

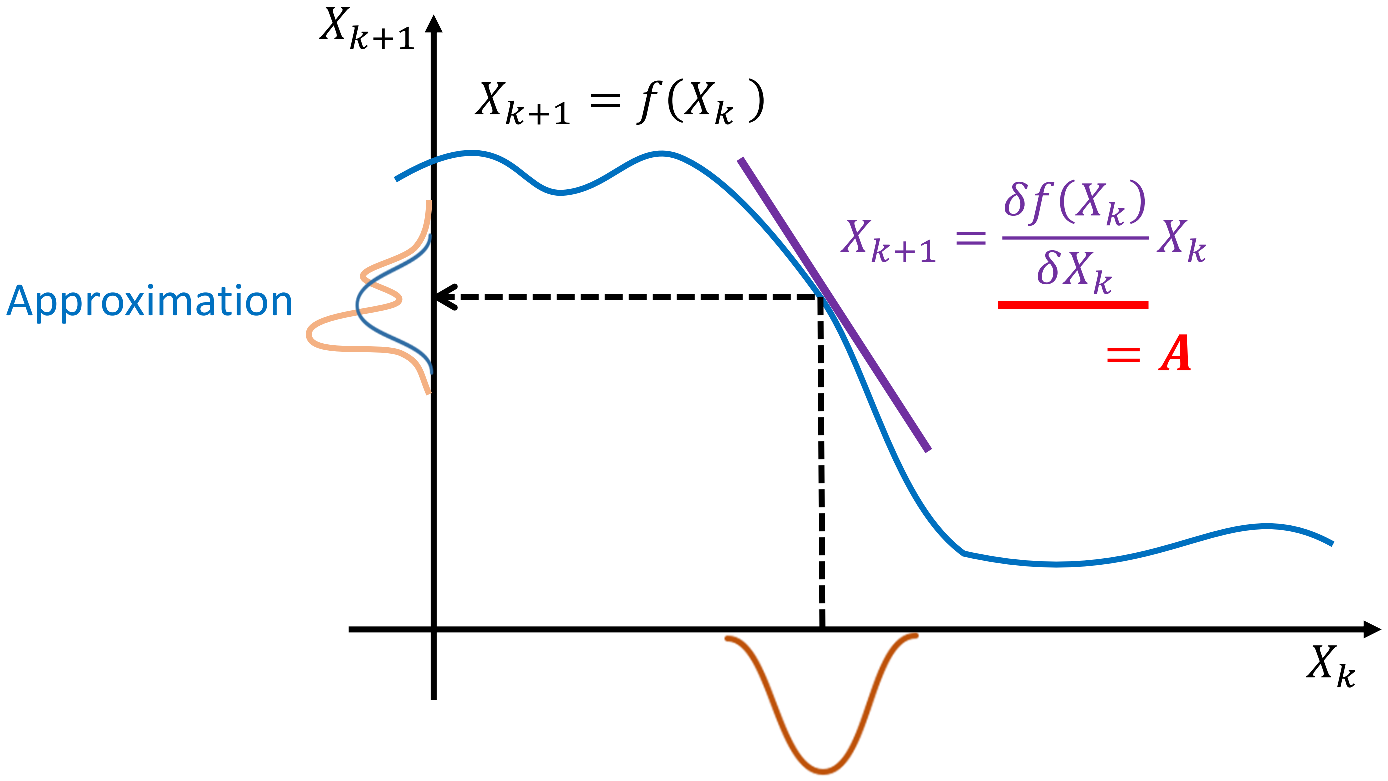
$$\dot{\psi}_{k+1} = \dot{\psi}_k = \boldsymbol{f_5(x_k, y_k, \dots)}$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ v_{k+1} \\ \psi_{k+1} \\ \dot{\psi}_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\delta f_1}{\delta x} & \frac{\delta f_1}{\delta y} & \frac{\delta f_1}{\delta v} & \frac{\delta f_1}{\delta \psi} & \frac{\delta f_1}{\delta \dot{\psi}} \\ \frac{\delta f_2}{\delta x} & \frac{\delta f_2}{\delta y} & \frac{\delta f_2}{\delta v} & \frac{\delta f_2}{\delta \psi} & \frac{\delta f_2}{\delta \dot{\psi}} \\ \frac{\delta f_3}{\delta x} & \frac{\delta f_3}{\delta y} & \frac{\delta f_3}{\delta v} & \frac{\delta f_3}{\delta \psi} & \frac{\delta f_3}{\delta \dot{\psi}} \\ \frac{\delta f_4}{\delta x} & \frac{\delta f_4}{\delta y} & \frac{\delta f_4}{\delta v} & \frac{\delta f_4}{\delta \psi} & \frac{\delta f_4}{\delta \dot{\psi}} \\ \frac{\delta f_5}{\delta x} & \frac{\delta f_5}{\delta y} & \frac{\delta f_5}{\delta v} & \frac{\delta f_5}{\delta \psi} & \frac{\delta f_5}{\delta \dot{\psi}} \end{bmatrix}}_{\text{Jacobian matrix}} \begin{bmatrix} x_k \\ y_k \\ v_k \\ \psi_k \\ \dot{\psi}_k \end{bmatrix} + \xi$$

$$= \frac{\delta f(X_k)}{\delta X_k}$$

Physics based prediction

- Nonlinear Kalman filter
  - Extended Kalman filter



① Prediction

$$\tilde{x}_k = A\hat{x}_{k-1} + Bu_k$$

$$\tilde{y}_k = C\tilde{x}_k$$

$$\tilde{P}_k = AP_{k-1}A^T + Q_k$$

② Correction

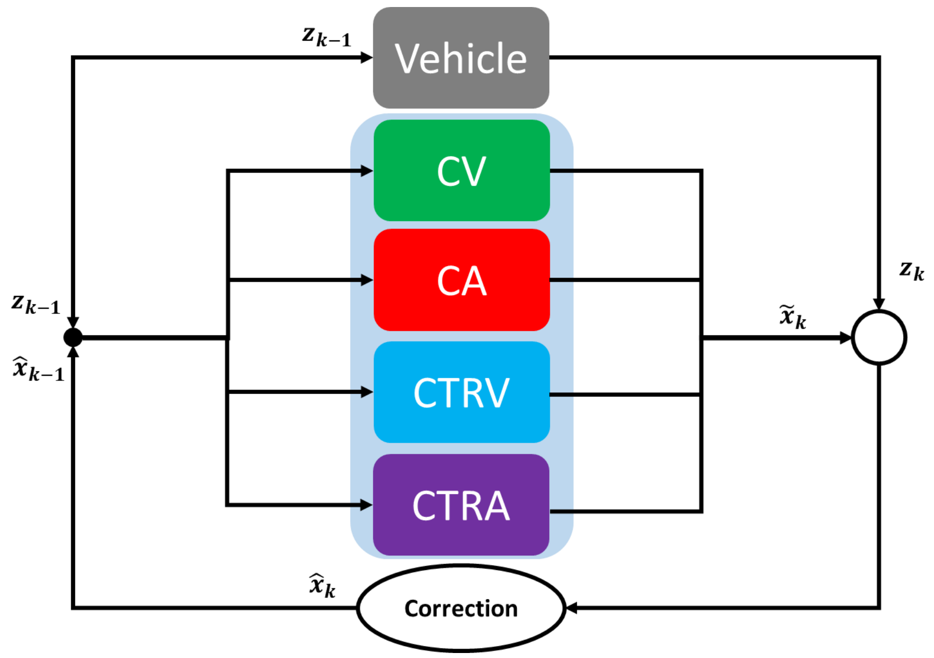
$$\hat{x}_k = \tilde{x}_k + K_k(y_k - \tilde{y}_k)$$

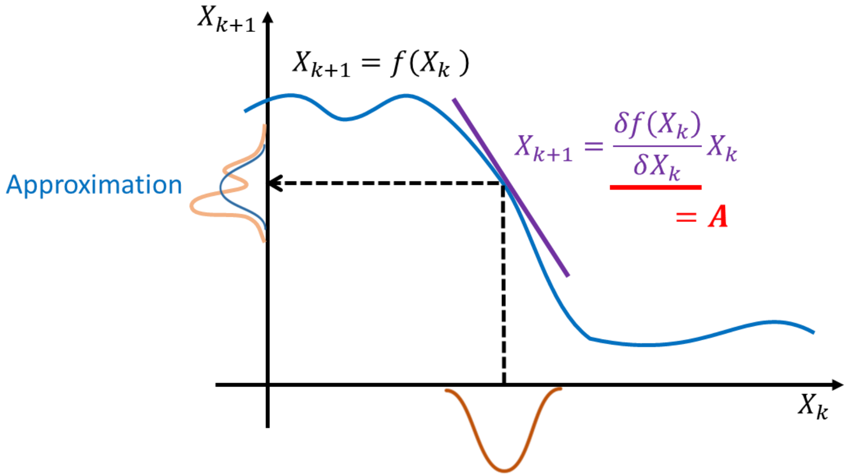
$$K_k = \frac{\tilde{P}_k C^T}{C\tilde{P}_k C^T + R_k}$$

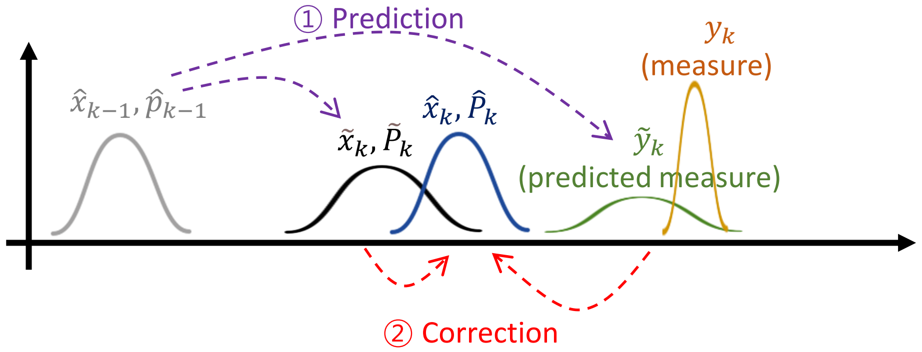
$$P_k = (I - K_k C)\tilde{P}_k$$

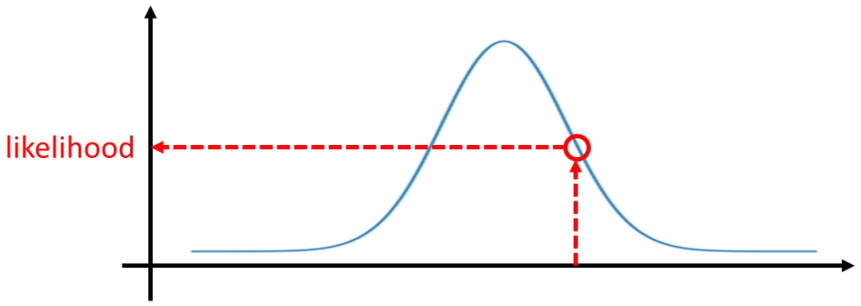
Physics based prediction

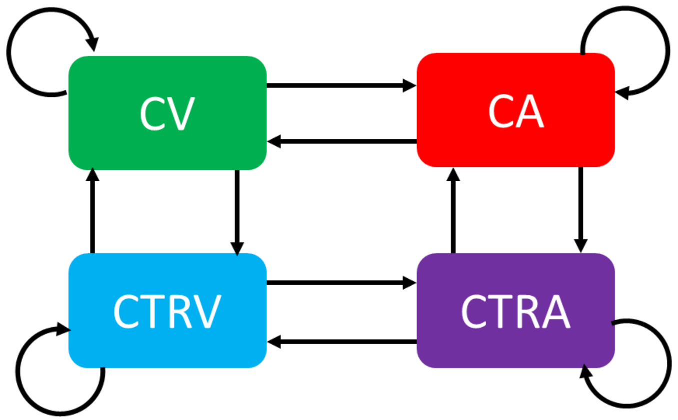
- Interacting multiple model (IMM)





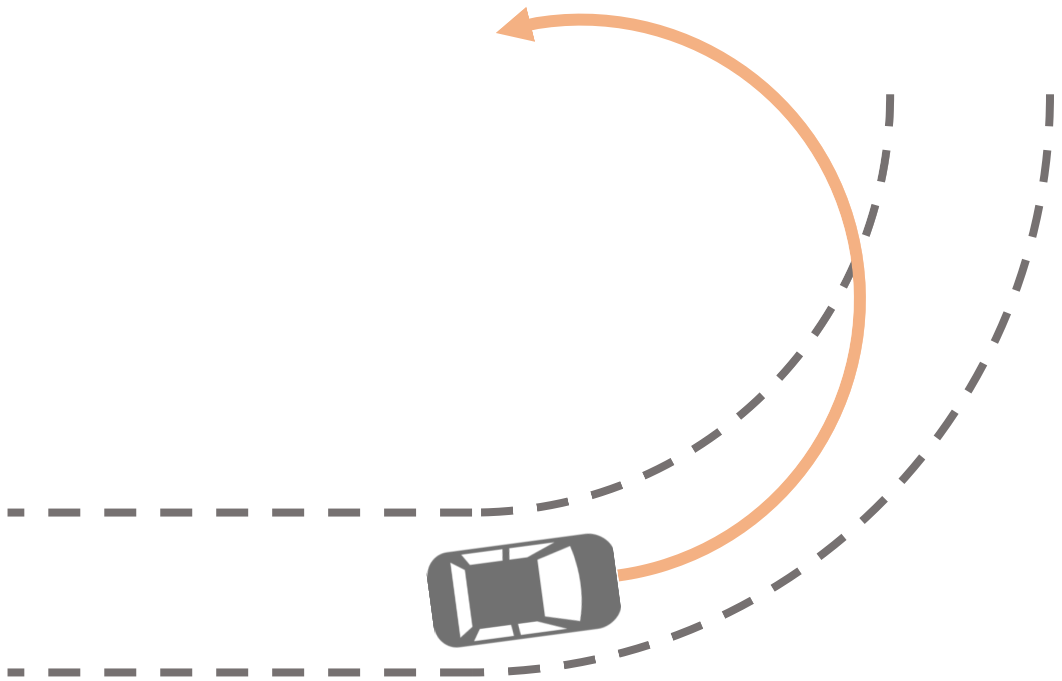


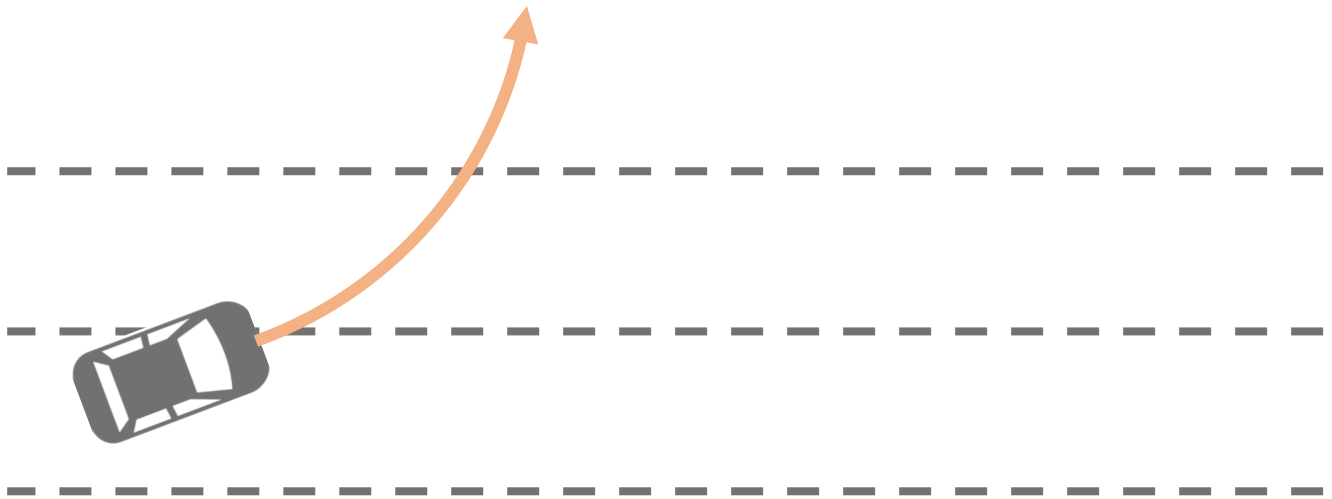




Physics based prediction

- Limits of physics-based prediction
  - Long-term prediction 에는 적합하지 않음
  - 도로 환경(차선 등) 을 고려할 수 없음
  - Frenet 환경에서 사용 가능?

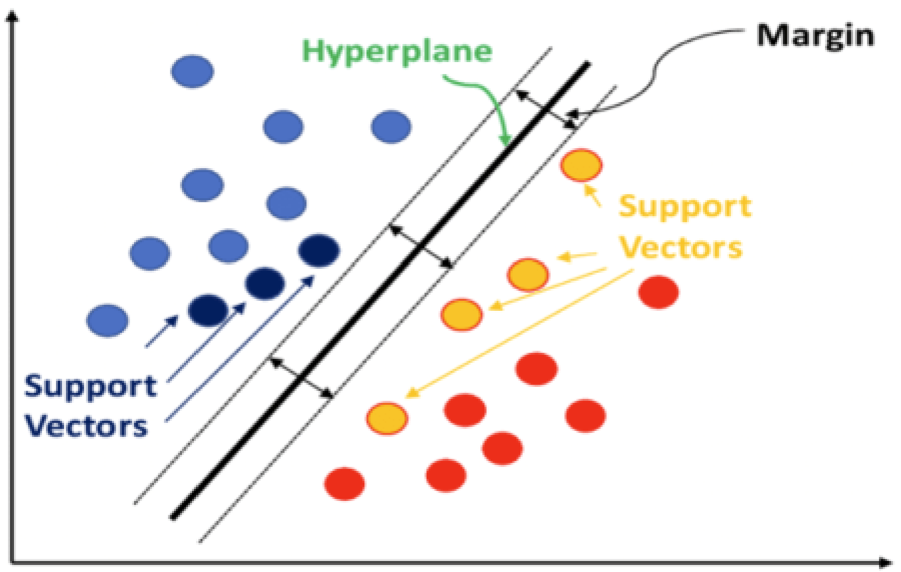


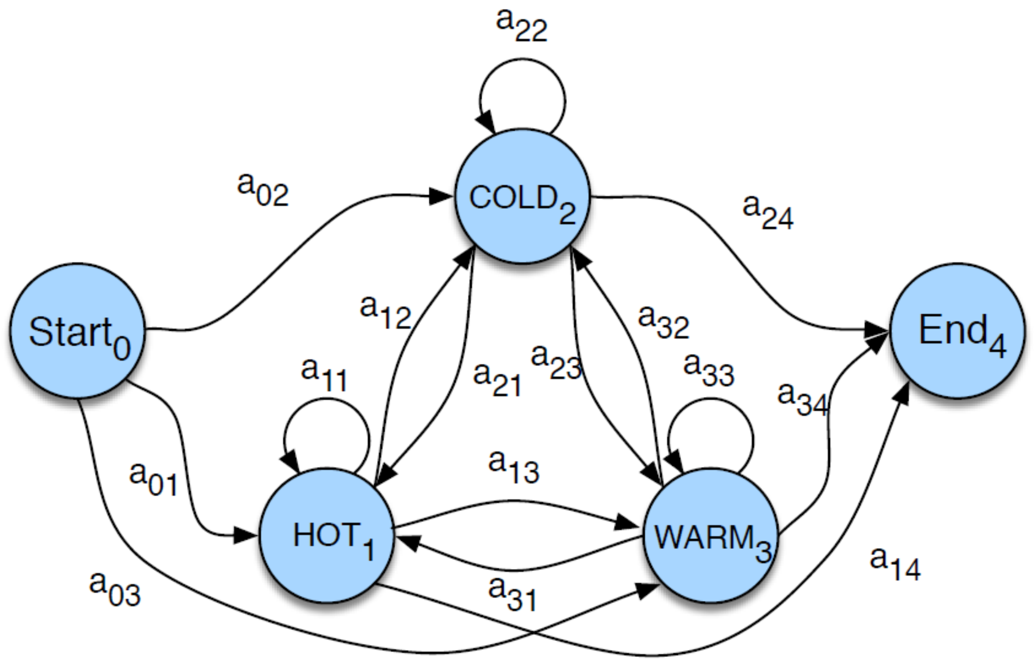


HL Klemove

Maneuver based prediction

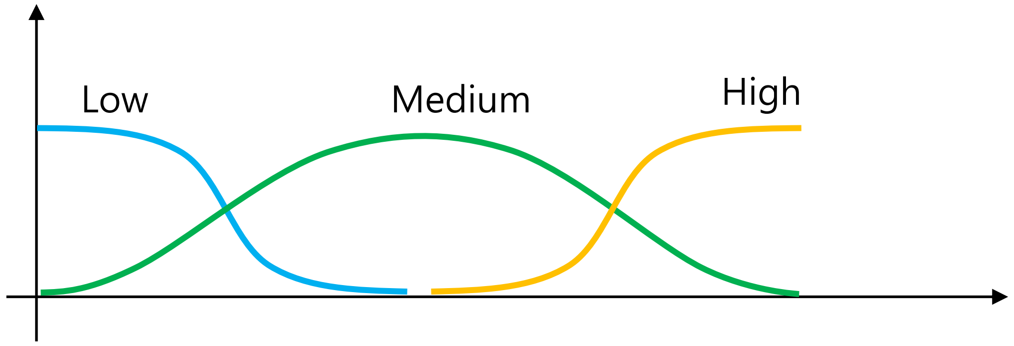
- Introduction
  - 다소 긴 시간 예측에 유리
  - 매우 긴 시간에 대한 예측은 불가
  - 주변 환경을 고려하도록 설계
  - Bayesian, SVM, HMM, Fuzzy, …





Maneuver based prediction

- Fuzzy theory
  - Fuzzification(퍼지화)
    - Membership function
      - Input : lateral velocity
      - Input label : Low / Medium / High lateral movement
      - Output label : Lane keeping / Lane changing
      - Form : sigmoid

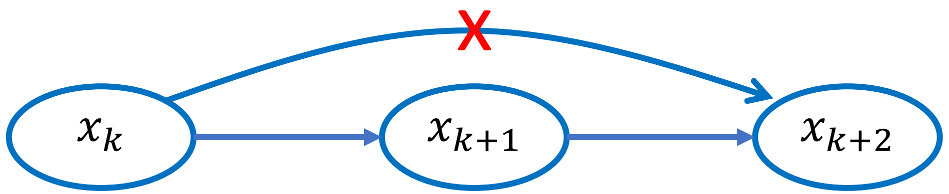


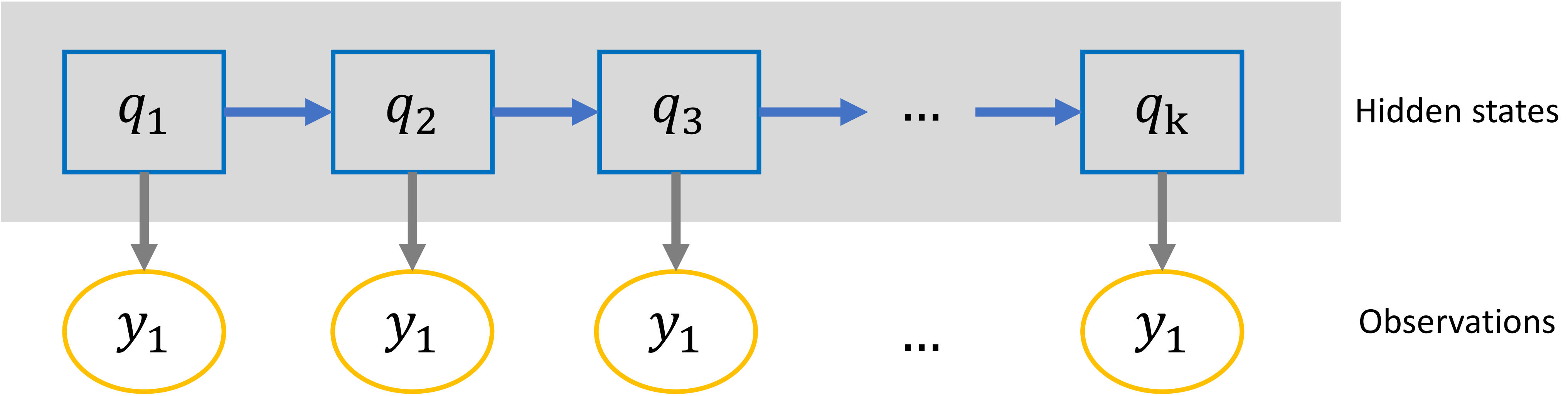
  - Fuzzy rules (퍼지 규칙)
    - If |v_lat| is low, the driver wants to keep lane for 100% chance
    - If | v_lat | is medium, the driver wants to keep lane for 50%, chang] lane for 50%
    - If | v_lat | is high, driver wants to change lane for 100% chance
  - Defuzzification (디퍼지화)
    - If the driver wants to keep lane for 100%, the vehicle will not start lane change within 10 s (0s~10s)
    - If the driver wants to change lane for 100%, the vehicle will finish lane change within 5s (10s~5s) after starting lane change

HL Klemove

Maneuver based prediction

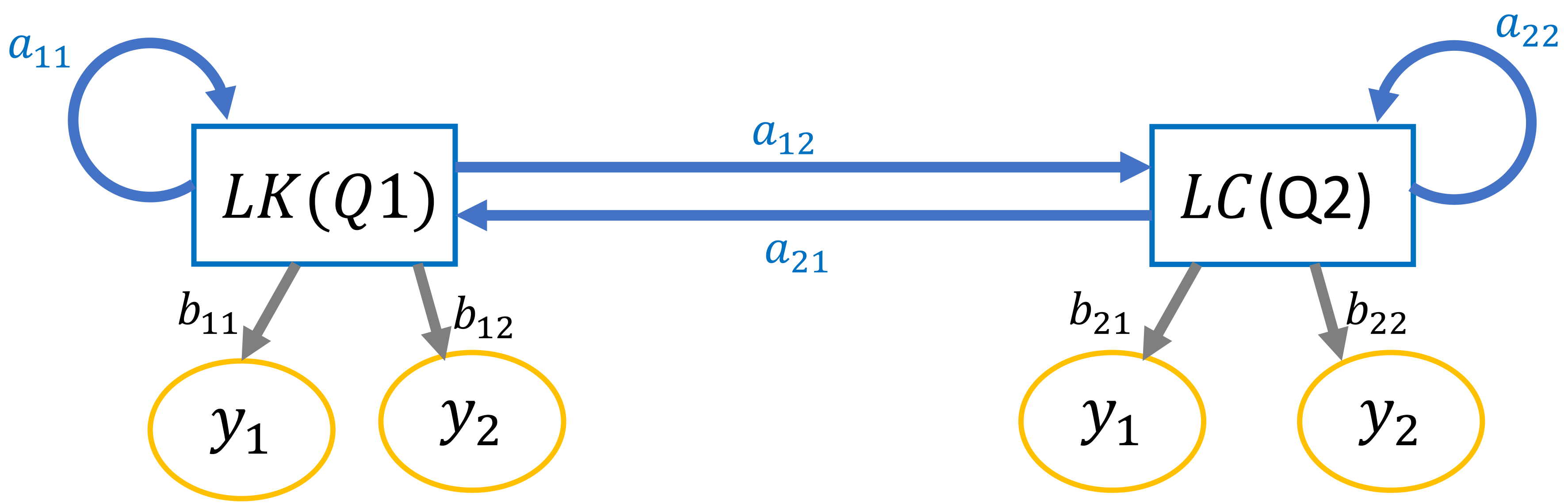
- Hidden Markov model
  - Main idea : 관측된 데이터는 숨겨진 State (Hidden state) 에서 비롯됨
  - Hidden state (Q) : State which we cannot observe. (Intension, Tendency, ⋯)
  - Observation (Y) : Value which we are able to observe. (Distance from centerline, acceleration, velocity, ⋯)
  - Assumption : Markov chain(model)





Maneuver based prediction

- Hidden Markov model
  - Transition matrix (A)
  - Emission matrix (B)
  - Initial probability (π)

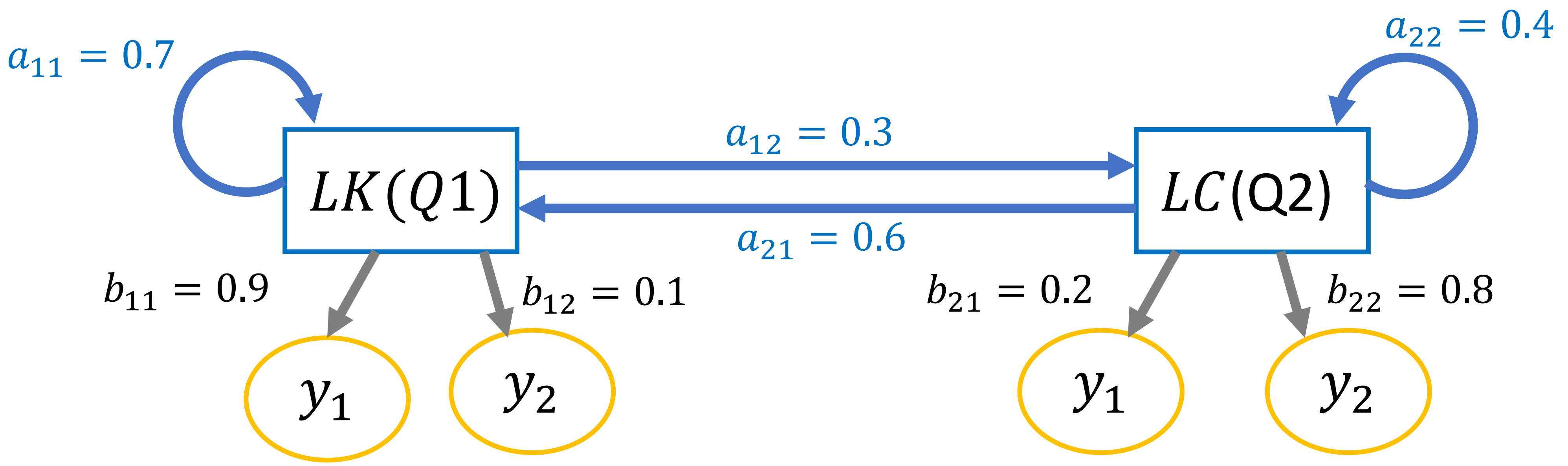


$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

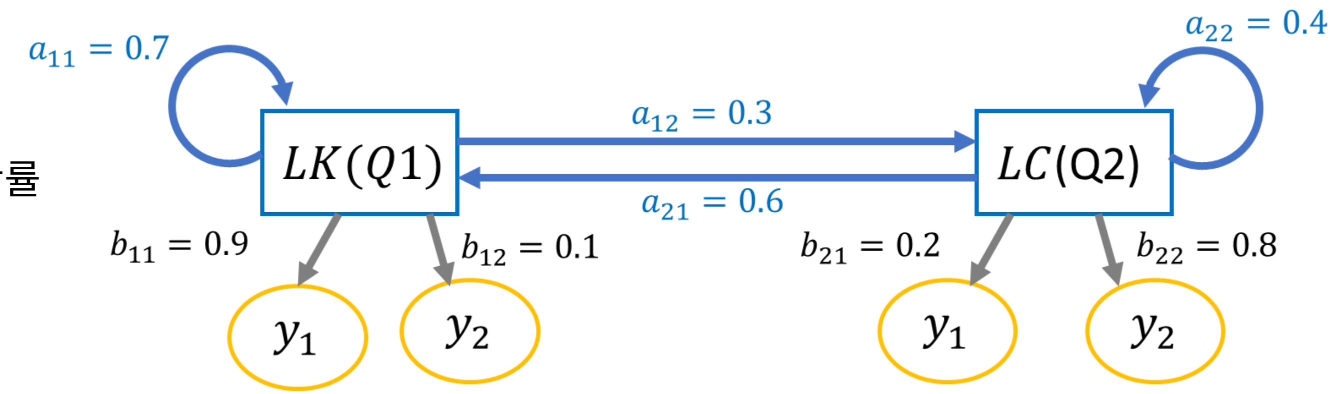
Maneuver based prediction

- Hidden Markov model
  - Example : P(y = {y1, y1, y2})



Maneuver based prediction

- Hidden Markov model
  - DP Problem : 반복되는 계산 값을 저장해서 사용
  - Forward probability (α) : 관측값이 주어졌을 때(1:k) 어떤 State(k)에 있을 확률



$$\alpha_{k,j} = \sum_q \alpha_{k-1} a_{qj}\, b_{jy}$$

$LK(Q1)$

$\alpha_1 = \pi_1 b_{11} = 0.9 \text{ x } 0.9 = 0.81$

LC(Q2)

$\alpha_2 = \pi_2 b_{21} = 0.1 \text{ x } 0.1 = 0.01$

y_1

$LK(Q1)$

$\alpha_1 = 0.81 \text{ x } 0.7 \text{ x } 0.1 + 0.01 \text{ x } 0.6 \text{ x } 0.1$

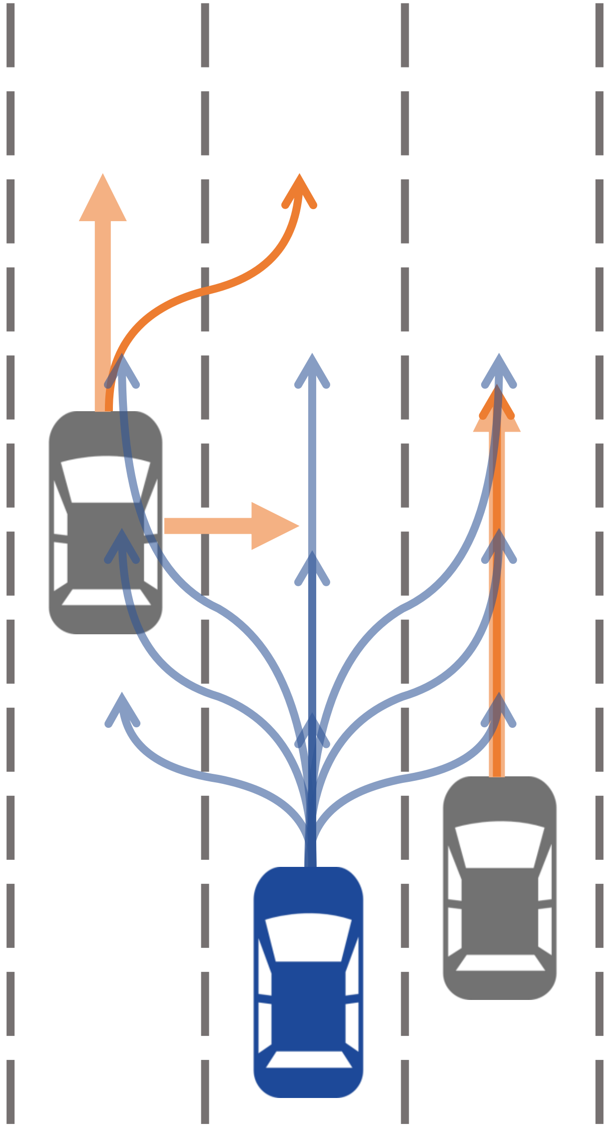
LC(Q2)

$\alpha_2 = 0.81 \text{ x } 0.3 \text{ x } 0.8 + 0.01 \text{ x } 0.4 \text{ x } 0.8$

y_2

Prediction

- Prediction on frenet frame
  - 선회 모델이 불필요
  - 모든 차들은 기본적으로 차선을 따라 주행한다!
  - 종방향 : Physical prediction
    - CV / CA 모델
  - 횡방향 : Maneuver prediction or physical prediction
    - Lane keeping / Lane change / CV / CA
  - 자차의 각 Trajectory candidate 별 Cost 계산 가능!



HL Klemove

HL Klemove

Thank You